(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 September 2005 (29.09.2005)

PCT

(10) International Publication Number
WO 2005/089241 A3

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:
PCT/US2005/008291

(22) International Filing Date: 11 March 2005 (11.03.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/552,653 13 March 2004 (13.03.2004) US

(71) Applicant *(for all designated States except US)*: **CLUSTER RESOURCES, INC.** [US/US]; 622 North 900 East, Spanish Fork, UT 84660 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **JACKSON, David, Brian** [US/US]; 10890 S. Conestoga Drive, Spanish Fork, UT 84660 (US).

(74) Agent: **ISAACSON, Thomas, M.**; The Law Office of Thomas M. Isaacson, 850 Lindy Lane, Huntingtown, MD 20639 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
— *of inventorship (Rule 4.17(iv))*

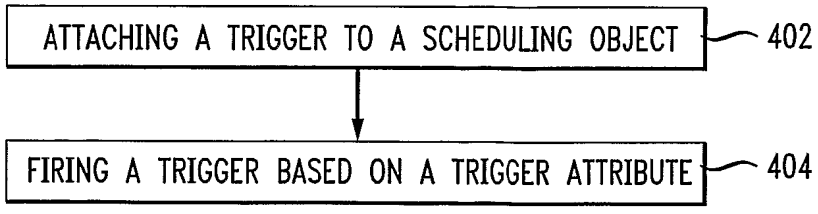
Published:
— *with international search report*
— *with amended claims*

(88) Date of publication of the international search report:
16 October 2008

Date of publication of the amended claims: 4 December 2008

(54) Title: SYSTEM AND METHOD FOR PROVIDING OBJECT TRIGGERS

FIG. 4



(57) Abstract: The present invention provides for systems and methods of dynamically controlling a cluster or grid environment. The method comprises attaching a trigger to an object and firing the trigger based on a trigger attribute. The cluster environment is modified by actions initiated when the trigger is fired. Each trigger has trigger attributes that govern when it is fired and actions it will take. The use of triggers enables a cluster environment to dynamically be modified with arbitrary actions to accommodate needs of arbitrary objects. Example objects include a compute node, compute resources, a cluster, groups of users, user credentials, jobs, resources managers, peer services and the like.

WO 2005/089241 A3

AMENDED CLAIMS
received by the International Bureau on 23 July 2008(23.07.2008)

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of dynamically controlling a compute environment, the method comprising:

attaching a trigger to a scheduling object in a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment; and

firing the trigger based on a trigger attribute, wherein the compute environment is modified by actions initiated when the trigger is fired.

2. (Original) The method of claim 1, wherein the compute environment is one of a compute farm, a cluster and a grid.

3. (Currently Amended) The method of claim 1, wherein modifying the compute environment comprises at least one of modifying interfaces, sending a notification, destroying or modifying [[the]] an object, adjusting policies, setting variables, creating files, executing scripts, provisioning nodes and utilizing a resource management service.

4. (Currently Amended) The method of claim 1, wherein the scheduling object is at least one of a job, a reservation for compute resources, a node, a user, a group of users, an administrator, an account, a QOS, a peer service, a class, a partition~~, a scheduling object~~ and a cluster.

5. (Original) The method of claim 1, wherein the trigger attribute is one of an event type, a timeout, an offset time, an action, a dependency, an action argument, a trigger state, an arbitrary input or output variable, a re-arm time, a time-out and a duration.

6. (Original) The method of claim 5, wherein if the trigger attribute is an event type, the event type is one of: the creation of the object, the starting, execution, modification, detection of failure, cancellation or termination of the object or an object state.

7. (Original) The method of claim 1, wherein modifying the compute environment when the trigger is fired further comprises utilizing data from the fired trigger to fire a second trigger.

8. (Original) The method of claim 7, wherein the data from the fired trigger relates setting variables causes the second trigger to fire.

9. (Original) The method of claim 7, wherein the data from the fired trigger can further cause a plurality of triggers to fire.

10. (Original) The method of claim 7, wherein the data relates to one of a threshold, a re-arm time, a time-out parameter and a duration.

11. (Original) The method of claim 1, wherein the action initiated when the trigger fires comprises obtaining data from a remote source.

12. (Original) The method of claim 1, wherein the remote source is the Internet.

13. (Original) The method of claim 1, wherein the action initiated when the trigger fires comprises updating a database.

14. (Original) The method of claim 1, wherein the action initiated when the trigger fires is based on a dynamic variable received via an argument list.

15. (Original) The method of claim 1, wherein the trigger attribute is a threshold value.

16. (Original) The method of claim 15, wherein the threshold value relates to one of: a quality of service, a queue time being exceeded and system performance.

17. (Original) The method of claim 16, wherein system performance is related to at least one of CPU load, disk activity, network usage and memory usage.

18. (Original) The method of claim 15, wherein the threshold value relates to one of: individual user parameters, group parameters, job parameters, administrative parameters and credentials.

19. (Currently Amended) A method of dynamically modifying a compute environment having at least one node, the method comprising:

attaching [[a]] an arbitrary and user-customizable trigger to a node within [[the]] a compute environment the compute environment comprising a plurality of commodity components running commodity software to process compute jobs submitted to the compute environment;

monitoring activities at the node via a node monitor;

upon detection of a specific activity at the node, firing the trigger to modify an attribute of the node.

20. (Original) The method of claim 19, wherein the attribute of the node comprises one of the node's priority and state.

21. (Original) The method of claim 19, wherein the node monitor monitors for local activity comprising at least one of mouse activity, keystrokes, and high levels of memory, network or CPU usage.

22. (Original) The method of claim 19, wherein firing the trigger to modify the node's attribute further comprises reducing a probability that the node will not be selected for a batch work load.

23. (Currently Amended) A method of dynamically modifying a compute environment having at least one node, the method comprising:

associating a trigger to at least one user of a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment;

monitoring activities of the at least one user associated with a threshold parameter; and

upon detecting that the monitored activities of the at least one user fail to pass the threshold parameter, firing the trigger to modify the compute environment to attempt to

enable the modified compute environment to accommodate the at least one user's activities at the threshold parameter.

24. (Original) The method of claim 23, wherein the threshold parameter relates to credentials of the at least one user.

25. (Original) The method of claim 23, wherein the threshold parameter relates to group credentials of the at least one user.

26. (Original) The method of claim 23, wherein if the threshold parameter is a performance threshold parameter.

27. (Original) The method of claim 23, further comprising, when the trigger fires, sending a notification to a compute environment administrator indicating the failure of the compute environment to pass the threshold parameter.

28. (Original) The method of claim 23, further comprising, when the trigger fires, adjusting a priority of jobs submitted by the at least one user to increase a probability that the at least one user's activities will pass the threshold parameter.

29. (Currently Amended) A system for dynamically controlling a compute environment, the system comprising:

a module configured to attach a trigger to an object in a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment; and

a module that fires the trigger based on a trigger attribute, wherein the compute environment is modified by actions initiated when the trigger is fired.

30. (Currently Amended) A system of dynamically modifying a compute environment having at least one node, the system comprising:

a module configured to attach a trigger to a node within [[the]] a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment;

a module configured to monitor activities at the node;

a module configured to fire the trigger to modify an attribute of the node upon detection of a specific activity at the node.

31. (Currently Amended) A computer-readable medium storing instructions for controlling a computing device to dynamically control a compute environment, the instructions comprising the steps:

attaching a trigger to an object in a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment; and

firing the trigger based on a trigger attribute, wherein the ~~cluster~~ compute environment is modified by actions initiated when the trigger is fired.

32. (Currently Amended) A computer-readable medium storing instructions for controlling a computing device to dynamically control a compute environment, the instructions comprising the steps:

attaching a trigger to a node within [[the]] a compute environment, the compute environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment;

monitoring activities at the node via a node monitor;

upon detection of a specific activity at the node, firing the trigger to modify an attribute of the node.

33. (Currently Amended) A computer-readable medium storing instructions for controlling a computing device to dynamically control a compute environment, the instructions comprising the steps:

associating a trigger to at least one user in a compute environment, the compute

environment comprising a plurality of commodity components running commodity software to process compute jobs in the compute environment;

monitoring activities at the at least one user associated with a threshold parameter; and

upon detecting that the monitored activities of the at least one user fail to pass the threshold parameter, firing the trigger to modify the compute environment to attempt to enable the modified compute environment to accommodate the at least one user's activities at the threshold parameter.